GenAug: Data Augmentation for Finetuning Text Generators

Steven Y. Feng,[*][1] **Varun Gangal,**[*][1] **Dongyeop Kang,**[2] **Teruko Mitamura,**[1] **Eduard Hovy**[1]

[1]Carnegie Mellon University

{syfeng,vgangal,teruko,hovy}@cs.cmu.edu

[2]University of California, Berkeley

dongyeopk@berkeley.edu

Abstract

In this paper, we investigate data augmentation for text generation, which we call *GenAug*. Text generation and language modeling are important tasks within natural language processing, and are especially challenging for low-data regimes. We propose and evaluate various augmentation methods, including some that incorporate external knowledge, for finetuning GPT-2 on a subset of Yelp Reviews. We also examine the relationship between the amount of augmentation and the quality of the generated text. We utilize several metrics that evaluate important aspects of the generated text including its diversity and fluency. Our experiments demonstrate that insertion of character-level synthetic noise and keyword replacement with hypernyms are effective augmentation methods, and that the quality of generations improves to a peak at approximately three times the amount of original data.

Method	Text
Original Review	got sick from the food . overpriced and the only decent thing was the bread pudding . wouldn't go back even if i was paid a million dollars to do so .
Synthetic Noise (10%)	got **seick** from the **fotod** . **overhpriced** and the only decent **ting** was the bread pudding . wouldn't go back even if i was paid a million dollars to do so .
Synonym Replacement (3 keywords)	got sick from the food . overpriced and the only decent thing was the **scratch pud** . wouldn't go back even if i was paid a **one thousand thousand** dollars to do so .
Hyponym Replacement (3 keywords)	got sick from the food . overpriced and the only decent thing was the **crescent roll corn pudding** . wouldn't go back even if i was paid a million **kiribati dollar** to do so .
Hypernym Replacement (3 keywords)	got sick from the food . overpriced and the only decent thing was the **baked goods dish** . wouldn't go back even if i was paid a **large integer** dollars to do so .
Random Insertion (10%)	got sick from the food **nauseous** . overpriced and the only decent thing was the bread pudding . wouldn't go back even if i was paid a million dollars **boodle** to do so .
Semantic Text Exchange (60% MRT)	got sick from the **coffee** . overpriced and the **food was good** . wouldn't **come back** if i was **in a long hand washing machine** .

Table 1: Example of a Yelp review and its variations using our augmentation methods. Changes are bolded.

1 Introduction

Text generation is an important but difficult task within natural language processing (NLP). A major goal is for dialogue agents to generate human-like text. The development of strong pretrained text generators like GPT-2 (Radford et al., 2019) has made it easier to perform generation for new domains or task specifications. These models are typically finetuned on downstream tasks such as classification; however, the first stage of their training is language modeling. Effective language models are important not only for generation but many NLP tasks.

In-domain examples are needed for finetuning. Otherwise, the generated text, though fluent English, will not faithfully imbibe domain properties such as the vocabulary preferred, domain shifts in word meaning, and domain distribution over properties such as sentiment. The learned language model will also poorly replicate the domain. However, many domains are low-data. These models do not have enough data to learn domain-specific aspects of the text, especially without sacrificing aspects such as its fluency and diversity.

One approach is with text data augmentation. There is constantly an increasing demand for large amounts of text data. Compared to fields such as computer vision, augmentation techniques for NLP are limited. Collecting and cleaning data manually requires time and effort. Also, certain domains do not have sufficient data available to begin with.

Prior work in text augmentation has focused on classification tasks, and there has been limited investigation for generation. A possible explanation is that generation is more complicated; rather than predicting the correct label, the text itself must be produced and should satisfy properties typical of human text such as being fluent, logical, and di-

---

[*] Equal contribution by the two authors.

Code: https://github.com/styfeng/GenAug

verse. Evaluation of the text is also more difficult.

In this work, we focus on data augmentation for text generation. We call this *GenAug*, and to the best of our knowledge, are the first to investigate it. We explore various augmentation methods such as semantic text exchange (STE) (Feng et al., 2019) and replacing keywords within examples from a small subset of the Yelp Reviews dataset (Yelp). See Table 1 for examples.[1] We also assess the impact of augmentation amount: from 1.5x to 4x the original amount of training data.

We evaluate the quality of generated text by GPT-2 after finetuning on our augmented data compared to the original data only. We illustrate that several augmentation methods improve the quality of the generations. We also show that the quality follows a trend with the augmentation amount: it increases until a peak and decreases thereafter. Overall, our major contributions can be summarized as follows:

- We propose *GenAug*, which is data augmentation specifically for text generation.
- We introduce and evaluate various augmentation methods for GenAug including inserting synthetic noise and integrating external knowledge through lexical databases for keyword replacement. We demonstrate that synthetic noise and replacement with hypernyms improve the quality of generations.[2]
- We investigate the effects of the augmentation amount and discover that performance improves until approximately three times the original training data, where all aspects of the generated text are noticeably improved upon.[2]
- We propose and use a mix of new and existing metrics for evaluating aspects of the text including its diversity, fluency, semantic content preservation, and sentiment consistency.[3]

2 Methodology

2.1 Model: GPT-2

We use OpenAI's GPT-2 (Radford et al., 2019), specifically its default pretrained model with 117M parameters. GPT-2 is a large transformer-based language model trained to predict the next word given previous words in a text. It is trained on *WebText* - a variety of internet data from sources such as Reddit, and has been shown to generate fluent text given different input prompts.

We choose this model as it is reasonably sized, frequently used as a pretrained text generator, and would thus benefit significantly from our experiments and analysis. We use HuggingFace's implementation of GPT-2 (Wolf et al., 2019).

2.2 Dataset: Yelp Reviews (YR)

The Yelp Reviews (YR) dataset contains user reviews on businesses. We choose YR as it differs substantially in domain from the "*WebText*" data used to train GPT-2, which consisted mainly of newswire and discussion forum threads. Unlike other review corpora such as SST-2 (Socher et al., 2013), YR contains long reviews with many sentences, making generation non-trivial.

We randomly select a small subset of YR for our experiments: a training split of 50K, validation split of 15K, and test split of 2K. This is approximately 1% of YR, replicating a low-data regime. We call this *Yelp-LR* or *YLR* (LR stands for low-resource). We include a proportion of reviews of each star rating equal to the proportions within YR to replicate the distribution of sentiment in YR.[4]

Finetuning GPT-2 on YLR represents the gold or baseline model. For each augmentation experiment, we combine YLR with our augmented data and finetune GPT-2 on this combination while using the same 15K validation and 2K test splits.

2.3 Text Augmentation Methods (AM)

We explore various augmentation methods (AM) to produce different versions of our training reviews[5] (see Table 1 for examples), and analyze their effects on GPT-2's generations. We split each review in half; a prompt and a continuation portion. We finetune GPT-2 on the entire reviews, but different AM are applied to either the prompt portion or entire review. We feed the prompt portion of test reviews as input to generate continuations.

2.3.1 Random Insertion, Deletion, & Swap

We experiment with random insertion, deletion, and swap (the *"Random Trio"*) on our entire reviews. Wei and Zou (2019) used these along with synonym replacement for text classification, and we investigate their performance for generation.

For each training example, we randomly swap the positions of two words, insert a random syn-

onym of a word that is not a stopword[6] into a random location, and remove a word, with α = 5% and 10% (5% and 10% of the words are changed). Hence, we produce six total variations per example.

2.3.2 Semantic Text Exchange (STE)

We investigate Semantic Text Exchange (STE) as introduced in Feng et al. (2019) on the entire reviews. STE adjusts text to fit the semantic context of a word/phrase called the replacement entity (RE). We use Feng et al. (2019)'s SMERTI-Transformer by training on a subset of YLR.[7] It inserts the RE into the text by replacing another entity, masks words similar to the replaced entity, and fills in these masks using a masked language model.

SMERTI is designed for shorter text due to the limited ability of the model to learn longer temporal dependencies.[8] We break each review into windows, and perform STE on each. Our augmentations are the concatenation of the semantically adjusted windows. For each window, a random RE is chosen. The candidates REs are 150 of the 200 most frequent nouns in SMERTI's training set.[9] We use masking rate thresholds (MRT) of 20%, 40%, and 60% , which represent the maximum proportion of the text that can be masked and replaced.

2.3.3 Synthetic Noise

We add character-level synthetic noise to the prompt portion of reviews. For every word, at every character, we perform a character insertion, deletion, or swapping of two side-by-side characters. The insertions are lowercase letters.

The three events have an equal chance of occurring at every character equal to one-third the overall level of noise. We ignore the first and last character of every word to more closely imitate natural noise and typos (Belinkov and Bisk, 2017). We produce 5%, 10%, and 15% noise variations per review. The noised prompt is combined with the original continuation to form the augmentations.

2.3.4 Keyword Replacement

We experiment with replacing keywords within entire reviews. We use RAKE (Rose et al., 2010) for keyword extraction. Candidate replacements are extracted from the lexical database WordNet (Miller, 1995). We replace up to three keywords for each review, resulting in a maximum of three augmentations for each review. Unlike STE, our goal is not to adjust the text's overall semantics.

The keywords replaced are ordered by their RAKE score (e.g. the probability of being a keyword) and replaced with words with the same overall part-of-speech (POS). We use the Stanford POS Tagger (Toutanova et al., 2003). Previous replacements are kept intact as further ones occur. There are three replacement methods:

1. **Synonyms Replacement (WN-Syns)** replaces each chosen keyword with a randomly chosen synonym of the same POS, preserving the text's semantics as much as possible.
2. **Hyponyms Replacement (WN-Hypos)** replaces each chosen keyword with a randomly chosen hyponym of the same POS that has more specific meaning. Words can have multiple hyponyms which differ semantically.
3. **Hypernyms Replacement (WN-Hypers)** replaces each chosen keyword with the closest (lowest) hypernym of the same POS that carries more broad and high-level meaning.

2.4 Text Augmentation Amounts

We also assess the impact of the amount of augmentation on the generated text. Specifically, **1.5x**, **2x**, **3x**, and **4x** the original amount of data (e.g. 4x refers to each example having three augmentations). We use a combination of synthetic noise, STE, and keyword replacement, each augmenting $\frac{1}{3}$ the YLR training examples (WN-Syns, Hypos, and Hypers each augment $\frac{1}{9}$).

2.5 Evaluation Metrics

We evaluate generated continuations using various metrics assessing major aspects of the text including its diversity, fluency, semantic content preservation, and sentiment consistency. Arguably the two most important are text fluency and diversity.[10]

2.5.1 Diversity

We pick a broad range of diversity measures for both intra- and inter-continuation diversity.[11]

1. SELF-BLEU (SBLEU) (ZHU ET AL., 2018), for a sample population S, measures the mean similarity of each sample to other samples. It is expressed as $E_{s \sim S}[BLEU(s, S - \{s\})]$,

---

[6]We use the stopwords list from Onix.

[7]See Section §3.2 for SMERTI training details.

[8]Feng et al. (2019) perform STE on text $\leq$ 20 words long.

[9]See Appendix §B for sliding window algorithm details.

[10]We do not use BLEU (Papineni et al., 2002) as we only have a single ground-truth continuation per review.

[11]We evaluate diversity on the generated continuations only (not concatenated with their corresponding prompts).

where $BLEU(h, R)$ is the BLEU-4 score of a hypothesis h measured against a set of references R. We measure the average SBLEU of every batch of 100 continuations per test prompt.[12] Lower SBLEU values represent higher inter-continuation diversity.

2. UNIQUE TRIGRAMS (UTR) (Tevet and Berant, 2020; Li et al., 2016) measures the ratio of unique to total trigrams in a population of generations. Higher UTR represents greater diversity. Since UTR is defined at the population level, it can assess the extent of cross-continuation repetition.

3. TYPE-TOKEN RATIO (TTR) is the ratio of unique to total tokens in a piece of text, and serves as a measure of intra-continuation diversity. The higher the TTR, the more varied the vocabulary in a continuation.

4. RARE-WORDS (RWORDS) (See et al., 2019) is defined by the following:

$$E_{s \sim S}[\sum_{w \in s} -\log \frac{n_{train}(w)}{N_{train}}]$$

where $n_{train}(w)$ and N_{train} are the corpus frequency of word w and the total corpus word count, respectively. Our corpus here is the 50K YLR training split. Lower values indicate usage of more rare words (less frequent in the corpus) and higher diversity.

2.5.2 Fluency

Fluency, also known as naturalness or readability, is a measure of how fluent text is. The higher the fluency, the more it imitates grammatically and logically correct human text.[13]

1. PERPLEXITY (PPL) is defined as:

$$PPL(S) = exp(-\frac{1}{|S|} ln(p_M(S)))$$

where S is a piece of text and $p_M(S)$ is the probability assigned to S by the language model. We finetune GPT-2 on a two-million review subset of YR (with a 500K additional validation split) and use this finetuned model for PPL evaluation. Outputs less likely to be seen in YR will typically have higher PPL.

2. SLOR (syntactic log-odds ratio) (Kann et al., 2018) is our main fluency metric. It modifies

PPL by normalizing for individual tokens (e.g. "Zimbabwe" is less frequent than "France" but just as fluent), and serves as a better measure. Higher SLOR represents higher fluency. The equation for SLOR is as follows:

$$SLOR(S) = \frac{1}{|S|}(ln(p_M(S)) - ln(\prod_{t \in S} p(t)))$$

where $|S|$ is the length of S (in tokens), $p_M(S)$ is the probability of S under language model M, and $p(t)$ are the unconditional probabilities of individual tokens (or unigrams) t in S. We use the same finetuned GPT-2 model on YR as for PPL mentioned above for SLOR. We use the proportional frequencies of unigrams in the two-million reviews as the unconditional unigram probabilities. Specifically, for tokens t: $p(t) = \frac{f(t)}{z+1}$, where $f(t)$ is the frequency of token t and $z = \sum_t f(t)$.

3. SPELLCHECK: For synthetic noise, we measure two spelling related metrics:

   (a) SPELLWORDS: average number of misspelled words per continuation.
   (b) SPELLCHARS: average number of character level mistakes per continuation.

These approximately measure how *noisy* the generations are, which can misleadingly improve diversity metrics. We use SymSpell (Garbe, 2019), which uses a Symmetric Delete Algorithm to quickly compute edit distances to a predefined dictionary. We set *verbosity* to *top*, a prefix length of ten, and consider a maximum edit distance of five.

2.5.3 Semantic Content Preservation (SCP)

SCP assesses how closely each generated continuation (hypothesis) matches in semantic content to the ground truth distribution of continuations (reference). Since the latter is unavailable in this case, we use the prompt itself as a proxy for reference.[14]

We use what we call the Prompt-Continuation BertScore (BPRO). BPRO computes average BertScore (Zhang et al., 2019a) between each continuation and the prompt. BertScore computes per-token BERT representations for both hypothesis and reference and aligns each hypothesis token to a reference token. We prefer BertScore over symbolic measures (e.g BLEU) since it does not rely on exact string matching alone and allows soft matches between different parts of the input pair.

---

[12]This is because we generate 100 continuations per test example. See Section §3.4 for more.

[13]We evaluate perplexity and SLOR on the concatenations of the generated continuations with their corresponding prompts, and Spellcheck on the generated continuations only.

[14]Datasets with multiple continuations per prompt are rare, and one continuation would be insufficient in most cases.

2.5.4 Sentiment Consistency

We finetune a BERT (Devlin et al., 2019) sentiment regressor on YLR by converting review stars into values between 0 and 1, inclusive, with higher values representing more positive sentiment.[15] We run the regressor on the ground-truth test reviews and the concatenation of our generated continuations with their corresponding prompts. We measure:

1. SENTSTD: average standard deviation of sentiment scores among each batch of 100 continuations (each concatenated with the input prompt) for a given test example. We do this for all 2000 test examples (100 prompt + continuation concatenations each) and take the average of the standard deviation values for each. A lower value indicates more consistent (lower spread) of sentiment, on average, among the continuations for each prompt.

2. SENTDIFF: average difference in sentiment score between each batch of 100 continuations (each concatenated with the single input prompt) and the corresponding ground-truth review in its entirety (essentially, the input prompt concatenated with the ground-truth continuation). We run this for all 2000 test examples (100 prompt + continuation concatenations each) and take the average of the differences. A lower value indicates sentiment of the continuations that, on average, more closely aligns with the ground-truth reviews.

3 Experiments

3.1 GPT-2 Finetuning

We finetune GPT-2 with a batch size of two. We try three different learning rates on YLR: 5e-4, 5e-5, and 5e-6, and find 5e-5 results in the lowest validation perplexity and use it for all experiments. We ensure the same hyperparameters and settings are used for each experiment. Final models correspond to epochs with the lowest validation perplexity.[16]

3.2 SMERTI-Transformer Training

We take a 25K subset of YLR's training split and a 7.5K subset of YLR's validation split. These serve as SMERTI's training and validation splits, respectively. This replicates the low-data regime, ensures SMERTI does not see additional data, and ensures SMERTI only learns from a portion of the data to prevent overfitting and repetition.

Each chosen example is split into chunks (or windows) of up to 30 tokens each,[17] resulting in 144.6K total training and 43.2K total validation examples for SMERTI. We mask 20%, 40%, and 60% of the words in $\frac{1}{3}$ of the examples each. We train SMERTI on this data and find the best performance after 9 epochs with a validation loss of 1.63. We use scaled dot-product attention and the same hyperparameters as Feng et al. (2019).[18]

3.3 Data Processing

For Yelp preprocessing, we filter out reviews that are blank, non-English, or contain URLs. For remaining ones, we remove repeated punctuations and uncommon symbols. For postprocessing, we noticed that many GPT-2 generations included trailing exclamation marks. We stripped these if more than four occurred in a row. Resulting blank continuations (very small portion of the total) were represented with a $<blank>$ token and ignored during evaluation of most metrics.

3.4 Experimental Setup

For the separate method experiments, we choose one augmentation for each training example, for a total of 2x the amount of original data. Since each method has multiple variations per training example, we randomly select one of these for each.

For the augmentation amount experiments, we ensure that larger amounts are supersets of smaller amounts - e.g. 3x contains all of the augmentation examples within 2x, and so forth.

We generate 100 continuations per test example by feeding in the prompt portions (first 50% of words). We use the default end-of-text token, a nucleus sampling budget (Holtzman et al., 2019) of 0.9, and a length limit of 500 for the generations.

For all experiments, we run two sets of random seeds, where each set $\{rs_1, rs_2\}$ consists of rs_1: a seed for data preparation and selection, and rs_2: a seed for GPT-2 finetuning and generation. Our final evaluation results are the average results.

4 Results and Analysis

4.1 Evaluation Results

Tables 2 and 3 contain average evaluation results for the variations and amounts, respectively. See Appendix §F for significance p-values and Ap-

---

[15]See Appendix §C for regressor finetuning details.

[16]See Appendix §D for details of the finetuned models.

[17]See Appendix §B for sliding window algorithm details.

[18]See Appendix §E for further SMERTI training details.

pendix §G for PPL results.[19] Figures 1 to 4 contain graphs of the variation results, and Figures 5 to 8 contain graphs of the amount results. The horizontal line(s) on the graphs refer to the no-augmentation (gold and 1x) setting with Yelp-LR. Table 4 contains generation examples.[20]
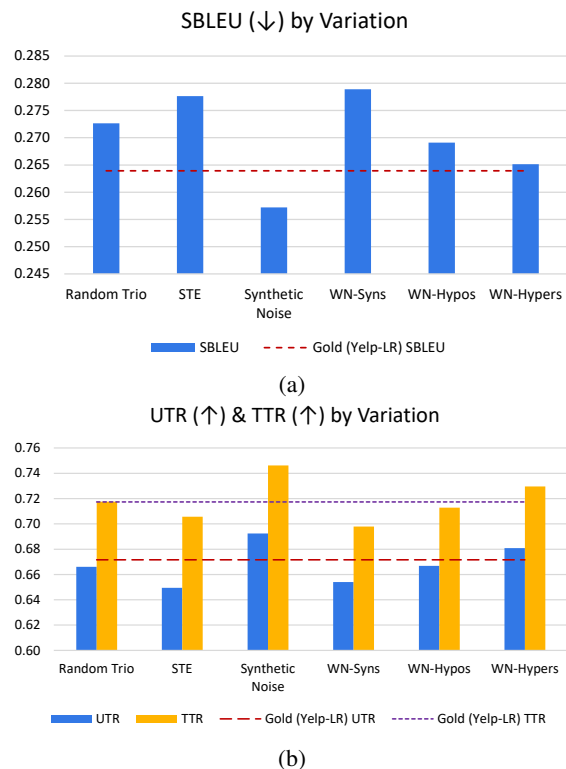
SBLEU (↓) by Variation

(a)

UTR (↑) & TTR (↑) by Variation

(b)

Figure 1: Graphs of a) average SBLEU and b) average UTR and TTR results by variation.

---

[19]Statistical significances are from paired two-tailed t-tests between the Yelp-LR and particular variation and amount results using an α of 0.05.
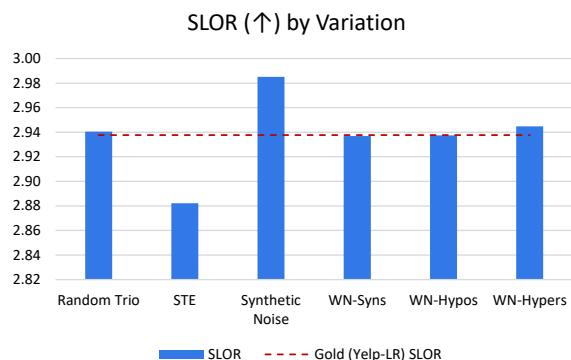
[20]See Appendix §H for more example generations.

SLOR (↑) by Variation

Figure 2: Graph of average SLOR results by variation.
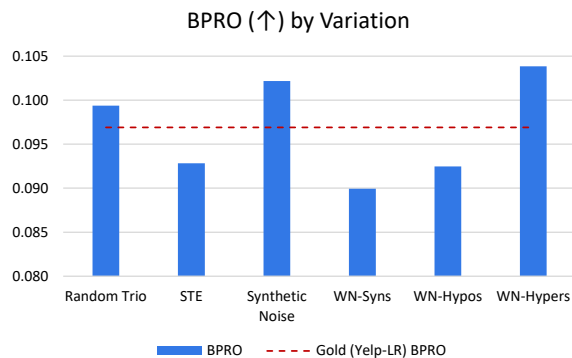
BPRO (↑) by Variation

Figure 3: Graph of average BPRO results by variation.

SentStd (↓) & SentDiff (↓) by Variation
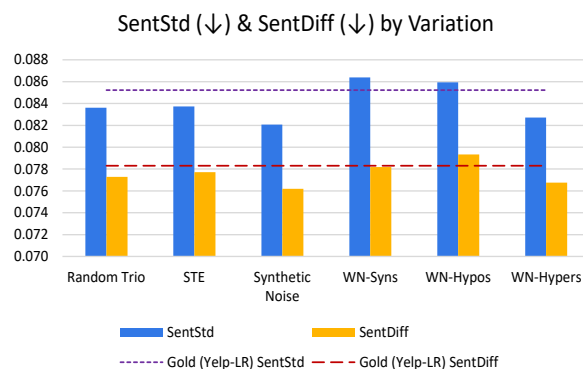

Figure 4: Graph of avg. sentiment results by variation.

4.2 Performance by Augmentation Method

We analyze the performance of each augmentation method using Table 2 and Figures 1 to 4.

4.2.1 Synthetic Noise and WN-Hypers

Synthetic Noise beats gold considerably on every metric. WN-Hypers does as well (other than SBLEU), but to a lesser extent on most metrics. Both clearly improve upon the gold setting.

To ensure Synthetic Noise's diversity improvements are not due to increased misspellings, we measure SpellWords and SpellChars. As seen in Table 5, Synthetic Noise actually decreases the average number of misspellings. This is likely because we only insert noise into the prompt portion of the training reviews, and GPT-2 is learning to be more robust to noise when finetuned on this data. This may also lead to increased generation quality.

WN-Hypers may improve performance as it slightly semantically adjusts the text. It does not keep semantics the same (unlike the goal of WN-Syns) but also does not drift too far since we choose the closest hypernyms. Each one carries more high-level meaning, which may contribute to increasing text diversity and fluency. We hence show that the

Variations	Gold (Yelp-LR)	Random Trio	STE	Synthetic Noise	WN-Syns	WN-Hypos	WN-Hypers
SBLEU (↓)	0.2639	0.2727	0.2776	**0.2572**	0.2789	0.2691	0.2651
UTR (↑)	0.6716	0.6660	0.6495	**0.6925**	0.6540	0.6669	**0.6808**
TTR (↑)	0.7173	**0.7176***	0.7056	**0.7461**	0.6978	0.7129	**0.7296**
RWords (↓)	-6.0637	**-6.0718**	-6.0508	**-6.1105**	**-6.0801**	**-6.0895**	**-6.0841**
SLOR (↑)	2.9377	**2.9404***	2.8822	**2.9851**	2.9368*	2.9373*	**2.9447**
BPRO (↑)	0.0969	**0.0994**	0.0928	**0.1022**	0.0899	0.0925	**0.1038**
SentStd (↓)	0.0852	**0.0836**	**0.0837**	**0.0821**	0.0864	0.0859*	**0.0827**
SentDiff (↓)	0.0783	**0.0773**	**0.0777***	**0.0762**	**0.0782***	0.0793	**0.0768**

Table 2: Average results by variation. Bold values indicate results better than Gold (Yelp-LR). Arrows beside each metric indicate whether lower or higher is better. * indicates insignificant values (using an α of 0.05).

Amounts	1x	1.5x	2x	3x	4x
SBLEU (↓)	0.2639	0.2724	0.2669	**0.2607**	**0.2583**
UTR (↑)	0.6716	0.6632	0.6678	**0.6837**	0.6707*
TTR (↑)	0.7173	0.7115	**0.7257**	**0.7535**	**0.7420**
RWords (↓)	-6.0637	**-6.0732**	**-6.0874**	**-6.1023**	**-6.0938**
SLOR (↑)	2.9377	**2.9435**	**2.9666**	**3.0001**	2.9258
BPRO (↑)	0.0969	**0.0971***	**0.1005**	**0.1067**	**0.0995**
SentStd (↓)	0.0852	**0.0840**	**0.0839**	**0.0784**	**0.0810**
SentDiff (↓)	0.0783	**0.0777***	**0.0775**	**0.0752**	**0.0771**

Table 3: Average results by amount. Bold values indicate results better than 1x (Yelp-LR). Arrows beside each metric indicate whether lower or higher is better. * indicates insignificant values (using an α of 0.05).

integration of external knowledge for GenAug can improve performance.

Unlike WN-Hypos, where replacements can be esoteric and rare, WN-Hypers' are typically more common words with higher chance of being seen by GPT-2 while training and appearing naturally at test-time. An example is replacing *dog* with *animal* (WN-Hypers) vs. *corgi* (WN-Hypos). Further, except quantified statements (e.g. *"All dogs bark"*), most WN-Hypers examples retain *faithfulness*[21] (Maynez et al., 2020) to the original, e.g. *"3 dogs walked home"* entails *"3 animals walked home"*.

4.2.2 STE and WN-Syns

STE and WN-Syns perform noticeably worse than gold. STE decreases fluency, diversity, and BPRO, albeit the sentiment-related metrics improve. WN-Syns decreases diversity and BPRO.

A possible explanation for STE is that SMERTI works best for shorter text.[22] Our sliding window is also problematic as text between windows may have semantic inconsistencies. For example, in Table 1, the chosen REs are *coffee* and *hand*; *hand*
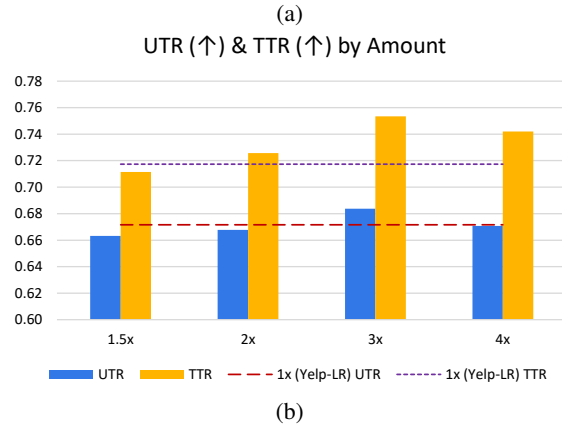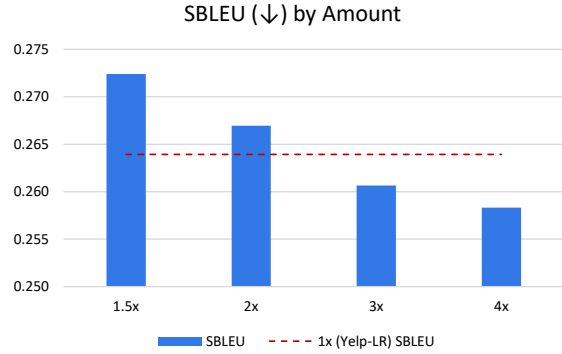

Figure 5: Graphs of a) average SBLEU and b) average UTR and TTR results by amount.

results in *washing machine*, making the last part semantically inconsistent with the first part about *coffee*. This likely results in reduced fluency and BPRO. Reduced fluency is also not unexpected as Feng et al. (2019) showed STE reduces SLOR.

A possible explanation for WN-Syns is that synonyms keep the semantic content almost exactly the same, unlike the other augmentation techniques which vary the semantics of the text more. Hence, GPT-2 may be overfitting to the augmented data.

4.2.3 WN-Hypos and Random Trio

Lastly, WN-Hypos and Random Trio also do not perform very well. WN-Hypos performs worse

Method	Text
Prompt	i got my hair and make up done here for my wedding on 12 29 13 . everything was amazing . hannah styled my hair and the results were pure perfection . i
Original	wish my hair could look like that everyday . i only have positive things to say about this place and would definitely recommend this place . i loved everything about this place !
Gold (Yelp-LR)	went home feeling amazing. you get a full set that changes throughout the year. thanks so much again hannah! you did an awesome job for me and my mom.
Synthetic Noise	am forever thankful for hannah and her store. she's been so nice and accommodating to my needs. she explained my wants and what i could do and she never backed off. i will definitely be back to her store. this is a terrific place for professional hair and make up
WN-Hypers	am so happy i came here and will absolutely continue coming here to get my perfect cut. i left well satisfied. i love this place! thanks yelpers and thank you hannah and make up artist anthony! you've earned my trust
2x	highly recommend this salon. they even have some coupons on their site. i also got my eyebrows and lip waxing here. very affordable too! i'll be back for sure
3x	couldn't believe how beautifully my hair turned out. my stylist was very quick and made sure to check on my hair every step of the way. the environment is a bit loud, but the receptionists and staff make up for it with a great quality of service and product. the price is right for the quality of the work. you'll definitely want to check this place out. i can't wait to return
4x	have to say i will definitely return to this salon. it's very romantic and upscale, all of the staff is very friendly and welcoming. i would definitely recommend this place to anyone who wants a beautiful hairdresser

Table 4: Examples of generated continuations from GPT-2 finetuned on select augmentation methods & amounts. *Prompt* is the first half of the original Yelp review fed in as input, and *Original* is the ground-truth continuation.
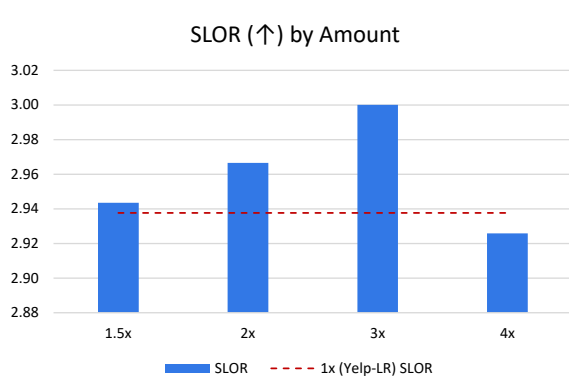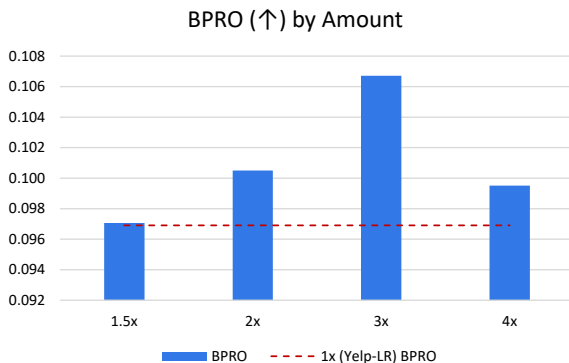

Figure 6: Graph of average SLOR results by amount.

Figure 8: Graph of avg. sentiment results by amount.

Figure 7: Graph of average BPRO results by amount.

Spellcheck	Gold (Yelp-LR)	Synthetic Noise
SpellWords (↓)	3.0024	**2.6274**
SpellChars (↓)	4.5804	**3.9190**

Table 5: Average Spellcheck results.

than gold on almost all metrics, but to a lesser extent. For Random Trio, overall diversity is decreased, but BPRO and sentiment-related metrics improve. A couple of Random Trio's metric improvements are minor and statistically insignificant.

This is likely due to Random Trio's techniques involving almost complete randomness (at the word-level), resulting in high variations in the metric results, leading to statistical insignificance and poor generations. Its random techniques appear much less suitable for GenAug than data augmentation for classification (Wei and Zou, 2019).

For WN-Hypos, we observe that some hyponyms diverge more from the parent than others (e.g. *food → beverage* vs. *food → micronutrient*), which can cause large drifts in meaning. Similar to Random Trio, this word-level *random-*

ness is likely leading to poor generations. Further, many hyponyms are esoteric words that GPT-2 has likely rarely (or never) seen (e.g *dragon→wyvern*), further decreasing performance. See the example in Table 1 (notice the word *kiribati*). Hence, we show that incorporation of external knowledge for GenAug can also decrease performance.

4.2.4 Overall Performance

Overall, Synthetic Noise and WN-Hypernyms are the best performing methods for GenAug on YLR (see Table 4 for example generations), and the others perform noticeably worse and are hence not recommended in their current state.

4.3 Performance by Augmentation Amount

Table 3 and Figures 5 to 8 show that quality of the generated text improves from 1.5x to 3x data augmentation, and decreases from 3x to 4x (except for SBLEU). 3x beats gold considerably on every metric, while 2x and 4x beat gold noticeably on most metrics as well (see Table 4 for example continuations). 1.5x performs noticeably worse than gold on text diversity.

Quality of the text really improves from 2x and onward, reaching a peak at 3x, and dropping afterward (especially in SLOR). For GenAug on YLR, 3x augmentation appears optimal, and more can reduce performance. This could be attributed to overfitting since many augmentation methods modify the original text to a limited degree. Augmentation at high amounts would thus have a similar (but lesser) effect to training on repeated examples.

5 Related Work

There has been work using GPT-2 as a component in the data augmentation process for training classifiers (Kumar et al., 2020; Papanikolaou and Pierleoni, 2020). We investigate augmentation for *finetuning GPT-2 itself*, and in fact deal with a precondition for the former - without a language model conforming to the domain, generated text would be further from the domain distribution.

There is also work on data augmentation for training NLP classifiers such as Wei and Zou (2019), Lu et al. (2006), and Kobayashi (2018). We adopt some techniques from Wei and Zou (2019) for our experiments, but in general, augmentation techniques for classification do not necessarily work well for generation. The distribution learned in the latter case, $P(x_c|x), x_c \in |V|^*$, is more complex than the former, $P(y|x), y \in Y \subset N$, due to a higher dimensional output variable (where Y is the label set, x_c denotes continuation, and $|V|$ refers to the vocabulary).

Generation of adversarial examples (AVEs) to evaluate robustness of NLP tasks is another area being investigated. Jia and Liang (2017) construct AVEs for span-based QA by adding sentences with distractor spans to passages. Zhang et al. (2019b) use word swapping to craft AVEs for paraphrase detection. Unlike these works, we are not concerned with test-time invariance or test-time model behavior on augmented examples, as long as these augmented examples improve training.

Kang et al. (2018) and Glockner et al. (2018) use WordNet relations to construct AVEs for textual entailment. However, to the best of our knowledge, we are the first ones to explore such methods using WordNet and lexical databases for text data augmentation for generative models.

6 Conclusion and Future Work

We introduced and investigated *GenAug*: data augmentation for text generation, specifically finetuning text generators, through various augmentation methods. We finetuned GPT-2 on a subset of the Yelp Reviews dataset, and demonstrated that insertion of character-level synthetic noise and keyword replacement with hypernyms are effective augmentation methods. We also showed that the quality of generated text improves to a peak at approximately three times the amount of original training data.

Potential future directions include exploring augmentation based on a) linguistic principles like compositionality (Andreas, 2020) and b) using more complex lexical resources - e.g. Framenet (Baker et al., 1998). One can also investigate further augmentation techniques using word replacement such as exploring the *contextual augmentation* method used in Kobayashi (2018). Further, methods of improving semantic text exchange (STE) on longer texts can be investigated, which would make it more effective for data augmentation. Lastly, there is potential in exploring data augmentation for other domains such as dialogue and related tasks such as style transfer (Kang et al., 2019), and investigating interesting aspects of it such as dialogue personalization (Li et al., 2020).

References

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Y. Feng, Aaron W. Li, and Jesse Hoey. 2019. Keep calm and switch on! preserving sentiment and fluency in semantic text exchange. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2701–2711, Hong Kong, China. Association for Computational Linguistics.

Wolf Garbe. 2019. Symspell. https://github.com/wolfgarbe/SymSpell.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 650–655.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Dongyeop Kang, Varun Gangal, and Eduard Hovy. 2019. (male, bachelor) and (female, Ph.D) have different connotations: Parallelly annotated stylistic language dataset with multiple personas. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1696–1706, Hong Kong, China. Association for Computational Linguistics.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. Adventure: Adversarial Training for Textual Entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2418–2428.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Aaron W Li, Veronica Jiang, Steven Y. Feng, Julia Sprague, Wei Zhou, and Jesse Hoey. 2020. Aloha: Artificial learning of human attributes for dialogue agents. In *Proceedings of Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 8155–8163.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xinghua Lu, Bin Zheng, Atulya Velivelli, and Chengxiang Zhai. 2006. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association : JAMIA*, 13:526–35.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Onix. Onix text retrieval toolkit stopword list 1. http://www.lextek.com/manuals/onix/stopwords1.html.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Guy Tevet and Jonathan Berant. 2020. Evaluating the evaluation of diversity in natural language generation. *arXiv preprint arXiv:2004.02990*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 173–180, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2078–2088, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yelp. Yelp open dataset. https://www.yelp.com/dataset.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research; Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

Appendices

A   Augmentation Variation Examples

See Tables 6 and 7 for further examples of Yelp review variations using our augmentation methods.

B   SMERTI Sliding Window Algorithm

We use 30-word windows, consisting of 10 words of context (the last 10 words of the previous window) and 20 new words.[23] In the context portion of each window, we cannot insert the RE nor mask or replace any words. In the new 20-word portion of each window, we can insert the new RE and mask and replace other words. This ensures when SMERTI performs STE on each window, it is able to utilize some context from the previous window but is unable to modify and blemish the STE already performed on the previous window.

C   Sentiment Regressor Finetuning

The BERT sentiment regressor is finetuned on the same Yelp-LR 50K training and 15K validation splits. The final classifer we use is after three epochs of finetuning. Details as follows:

- Star rating conversion: 1 star = 0, 2 star = 0.25, 3 star = 0.5, 4 star = 0.75, 5 star = 1
- Finetuning details:
    - max_seq_length: 128
    - per_gpu_eval_batch_size: 32
    - per_gpu_train_batch_size: 32
    - learning_rate: 2e-5

D   Finetuned Model Details

Note: BE below stands for "best epoch", and VPPL for "validation perplexity".

- Two-million review subset of Yelp (for PPL and SLOR eval): BE = 4, VPPL = 9.1588
- Seed set 1 finetuned models:
    - gpt2_gold: BE = 3, VPPL = 11.7309
    - gpt2_noise: BE = 3, VPPL = 12.0408
    - gpt2_STE: BE = 3, VPPL = 12.1892
    - gpt2_syns: BE = 2, VPPL = 11.9844
    - gpt2_hypos: BE = 2, VPPL = 11.9638
    - gpt2_hypers: BE = 2, VPPL = 12.0131
    - gpt2_random: BE = 2, VPPL = 11.9297
    - gpt2_1.5x: BE = 3, VPPL = 11.8958
    - gpt2_2x: BE = 3, VPPL = 11.9113

---

[23]The first window is 20 words long and has no context. If a review is at most 25 words long, we perform STE on the entire review (without the sliding window algorithm).

- gpt2_3x: BE = 2, VPPL = 12.2064
- gpt2_4x: BE = 1, VPPL = 12.3574
- Seed set 2 finetuned models:
    - gpt2_gold: BE = 3, VPPL = 11.7387
    - gpt2_noise: BE = 2, VPPL = 12.0230
    - gpt2_STE: BE = 3, VPPL = 12.1711
    - gpt2_syns: BE = 2, VPPL = 11.9282
    - gpt2_hypos: BE = 2, VPPL = 11.9583
    - gpt2_hypers: BE = 2, VPPL = 11.9957
    - gpt2_random: BE = 2, VPPL = 11.9558
    - gpt2_1.5x: BE = 3, VPPL = 11.8943
    - gpt2_2x: BE = 2, VPPL = 12.0209
    - gpt2_3x: BE = 2, VPPL = 12.1710
    - gpt2_4x: BE = 1, VPPL = 12.3288

E   SMERTI-Transformer Training

Similar to Feng et al. (2019), we use scaled dot-product attention and the same hyperparameters as Vaswani et al. (2017). We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We increase the learning rate (LR) linearly for the first $warmup\_steps$ training steps, and then decrease the LR proportionally to the inverse square root of the step number. We set $factor = 1$, $warmup\_steps = 2000$, and use a batch size of 4096.

F   Statistical Significance p-values

See Tables 8 and 10 for p-values of results by variation and amount, respectively. These are the results from paired two-tailed t-tests against Yelp-LR (Gold and 1x) results. We test statistical significance of all metrics other than RWords and PPL, and use an alpha of 0.05.

G   Perplexity (PPL) Results

See Tables 9 and 11 for average PPL results by variation and amount, respectively. Synthetic Noise, 2x, and 3x beat gold (Yelp-LR), similar to SLOR. However, WN-Hypers has higher PPL than gold (unlike SLOR). This is likely due to WN-Hypers having outputs that contain rarer tokens, thus increasing PPL. We note again that SLOR normalizes for this and is a better measure of fluency overall.

H   Generated Continuation Examples

See Tables 12 and 13 for further examples of generated continuations from the various experiments.

Method	Text
Original Review	fantastic selection of wines and always served at the proper temperature . the ambiance is stellar dark and cool like a wine cellar and the bands that i have seen there have been very good . check out their jazz band on monday night .
Synthetic Noise (15%)	fantastic **selectoin** of wines and always **sevred** at the **prouper temperaure** . the **ambfiaynce** is **sftellar dak** and cool like a wine cellar and the bands that i have seen there have been very good . check out their jazz band on monday night .
Synonym Replacement (3 keywords)	**wondrous option** of wines and always served at the **right** temperature . the ambiance is stellar dark and cool like a wine cellar and the bands that i have seen there have been very good . check out their jazz band on monday night .
Hyponym Replacement (3 keywords)	fantastic **write-in** of wines and always served at the proper **melting point** . the ambiance is stellar **gloom** and cool like a wine cellar and the bands that i have seen there have been very good . check out their jazz band on monday night .
Hypernym Replacement (3 keywords)	fantastic **action** of wines and always served at the proper **fundamental quantity** . the ambiance is stellar **illumination** and cool like a wine cellar and the bands that i have seen there have been very good . check out their jazz band on monday night .
Random Swap (10%)	fantastic selection of **cool** and always served at the proper temperature . the ambiance **i** stellar dark and **wines** like a wine cellar and **out** bands that **have** have seen there **is** been very good . check **the** their jazz band on monday night .
Semantic Text Exchange (60% MRT)	fantastic selection of wines and always served at the **same meat** . the **food is always fresh** and **the service is always friendly** and i have **to say** there have been very good . **they are out of the deal .**

Table 6: Example of a Yelp review and its variations using our augmentation methods. Changes are bolded.

Method	Text
Original Review	the girls working were so nice . they set up a table for us and gave honest , helpful opinions about the food . adorable store too ! great experience overall . we loved the breakfast sandwich .
Synthetic Noise (5%)	the girls working **wre** so **nwice** . they set up a table for us **ad** gave honest , **hlpful** opinions about the food . adorable store too ! great experience overall . we loved the breakfast sandwich .
Synonym Replacement (3 keywords)	the girls working were so nice . they set up a table for us and gave honest , helpful **view** about the food . **lovely storage** too ! great experience overall . we loved the breakfast sandwich .
Hyponym Replacement (3 keywords)	the girls working were so nice . they set up a table for us and gave honest , helpful **conclusion** about the food . adorable **beauty parlor** too ! great **familiarization** overall . we loved the breakfast sandwich .
Hypernym Replacement (3 keywords)	the girls working were so nice . they set up a table for us and gave honest , helpful **belief** about the food . adorable **mercantile establishment** too ! great **education** overall . we loved the breakfast sandwich .
Random Deletion (10%)	the girls working were so nice . they set up a table for us and gave honest , helpful opinions about food . adorable too ! experience overall . we the breakfast sandwich .
Semantic Text Exchange (60% MRT)	the **guys** working were **very** nice . they set up a **set** for us and gave **us a good time** , **very fun and fun and fun and fun fun with the ingredients** . adorable store too ! great experience overall . **we will definitely return .**

Table 7: Example of a Yelp review and its variations using our augmentation methods. Changes are bolded (except for Random Deletion where words were removed).

Variations	Random Trio	STE	Synthetic Noise	WN-Syns	WN-Hypos	WN-Hypers
SBLEU	4.288E-104	1.863E-164	5.521E-51	4.324E-283	1.164E-33	0.0018
UTR	4.566E-37	6.497E-261	0.0000	1.698E-288	5.311E-23	5.630E-102
TTR	0.6104*	2.390E-92	0.0000	2.358E-288	2.589E-16	2.149E-135
SLOR	0.1346*	2.694E-108	7.820E-117	0.6114*	0.8618*	0.0001
BPRO	1.071E-15	7.136E-31	2.828E-39	7.113E-94	3.217E-42	1.866E-100
SentStd	3.393E-05	0.0001	8.833E-14	0.0029	0.0570*	1.267E-11
SentDiff	0.0017	0.0709*	1.932E-08	0.7293*	0.0010	9.370E-08
SpellWords	N/A	N/A	0.0000	N/A	N/A	N/A
SpellChars	N/A	N/A	0.0000	N/A	N/A	N/A

Table 8: p-values of results by variation. Note: * indicates insignificant values (using an alpha of 0.05).

Variations	Gold (Yelp-LR)	Random Trio	STE	Synthetic Noise	WN-Syns	WN-Hypos	WN-Hypers
Perplexity ($\downarrow$)	71.9447	72.5887	76.6056	**71.1775**	73.2042	73.6881	77.7176

Table 9: Average perplexity results by variation. Note: bold values are better (lower) than gold (Yelp-LR).

Amounts	1.5x	2x	3x	4x
SBLEU	1.457E-98	1.001E-13	3.879E-12	1.112E-27
UTR	6.668E-71	4.738E-14	1.082E-84	0.2463*
TTR	7.868E-32	5.653E-65	0.0000	1.432E-295
SLOR	0.0020	3.404E-46	4.057E-151	7.017E-06
BPRO	0.6436*	1.571E-28	9.270E-121	7.356E-11
SentStd	0.0012	0.0004	3.190E-64	1.137E-23
SentDiff	0.0632*	0.0105	2.567E-18	0.0030

Table 10: p-values of results by amount. Note: * indicates insignificant values (using an alpha of 0.05).

Amounts	1x (Yelp-LR)	1.5x	2x	3x	4x
Perplexity ($\downarrow$)	71.9447	74.5190	**69.8858**	**67.2170**	74.9576

Table 11: Average perplexity results by amount. Note: bold values are better (lower) than 1x (Yelp-LR).

Method	Text
Prompt	the veal piccata was exquisite and my husband thoroughly enjoyed lobster ravioli . shared the house salad and it was fresh and
Original	plenty for two . thank you for our complimentary desert , it was the perfect finish . staff was awesome as always .
Gold (Yelp-LR)	excellent! also the baked dates were good, made to order. our server was attentive and friendly. i'm sure it will be back soon
Synthetic Noise	delicious. lobster ravioli was also excellent, crispy and savory. our waiter was attentive and had a funny attitude. the potatoes we ordered were absolutely fabulous. everything on the plate was perfection and very well presented. we will be back for sure.
WN-Hypers	tasty. portion size was perfect and meal was exactly how we were looking for. prices were a tad high for the amount of food but the quality was top notch. we will be back.
2x	tasty and decadent! also enjoyed the stuffed crab cakes, they were incredibly fresh and absolutely the perfect combination of creamy, flaky goodness! highly recommend this amazing restaurant! if you are looking for some unique and wonderful, fantastic meals, then come here and you will not be disappointed! great way to end an otherwise boring evening.
3x	the dressing was yummy. the only suggestion i would make is the price of the ravioli is a bit high. but, if you are looking for something light and hearty, this is the place to go. you could pay about $ 20 or so for a meal for two people
4x	delicious. excellent wine selection and the atmosphere was fine and classy. only downside was the parking. otherwise, a great place to hang out and eat with your family and friends.

Table 12: Examples of generated continuations from GPT-2 finetuned on select augmentation methods & amounts. *Prompt* is the first half of the original Yelp review fed in as input, and *Original* is the ground-truth continuation.

Method	Text
Prompt	good to very good gym . some of the machines need to be moved out to make room for newer
Original	more useful equip . need another smith machine def and poss another flat bench . and please add some hammer strength
Gold (Yelp-LR)	machines. good food selection. good staff, reasonable prices, and a quiet atmosphere. gym was clean and well kept, and the associate helped me check out the yoga area to help me decide on a new class i would take. i would highly recommend
Synthetic Noise	equipment. it's also very bright and well maintained. the gym is in a really convenient location. i would highly recommend if you like to workout at a good price.
WN-Hypers	machines. the men's room is a good size. i liked that there's a separate swimming pool area to cool off and dip in the pool. i would visit again for sure.
2x	machines and the trainers there was also quite impressive but the ac didn't work so i stopped going.
3x	machines. staff is pretty awesome. clean facility with lots of cardio equipment. after some recent renovations, it's nice to have a nice place to do work. pricing is very reasonable.
4x	equipment. the equipment is expected and the fitness staff was friendly and knowledgeable. it's a nice facility. i would recommend.

Table 13: Examples of generated continuations from GPT-2 finetuned on select augmentation methods & amounts. *Prompt* is the first half of the original Yelp review fed in as input, and *Original* is the ground-truth continuation.